Учетные записи пользователей и групп

1. Пользовательские учетные записи
2. БД паролей пользователей
3. Учетные записи групп
4. Информация по пользователям в системе
5. Повышение привелегий в системе

Раздел 1

Пользовательские учетные записи

Определения

- **Учетная запись** - Ключевой элемент системы разграничения доступа ОС. Пользователь, выполнив аутентификацию, представляется в системе в виде учетной записи. Идентификатор учетной записи пользователя, а также список групп, в которые пользователь входит становятся атрибутами запускаемых пользователем процессов, тем самым представляя пользователя в системе и определяя для процессов их разрешения доступа.

- **Доступ к ресурсам** - ОС предоставляет или отказывает процессу в **доступе к ресурсам** исходя из пользовательских идентификаторов процесса
- **Домашний каталог** - УЗ связаны с **домашним каталогом** пользователя и **“профилем”** - группой индивидуальных настоечных файлов, обеспечивая индивидуальное приватное пространство для работы пользователя

Именование пользователей

- первый символ - подчеркивание или латинская строчная буква a-z
- дальнейшие символы - цифры, латинские буквы, подчеркивание или дефис
- последний символ - не должен быть дефисом
- длина имени - поддерживаются идентификаторы до 32 символов

Иерархия УЗ пользователей

- **Суперпользователь (UID=0)** - учетная запись с нулевым идентификатором, обладает абсолютными полномочиями в системе
- **Служебные** учётные записи (0 < UID < UID_MIN) - учетные записи, предназначенные для работы служб и прочих системных процессов, не предполагают возможности интерактивного входа в систему
- **Операторские** учётные записи (UID_MIN <= UID < UID_MAX) - учетные записи обычных пользователей.

БД учетных записей - /etc/passwd

Каждая строчка - запись о пользователе, поля записи разделяются двоеточием

```
login : password : UID : GID : GECOS : home : shell
```

Поля /etc/passwd

```
beav:x:1000:1000:Theodore Cleaver:/home/beav:/bin/bash
warden:x:1001:1001:Ward Cleaver:/home/warden:/bin/bash
dobie:x:1002:1002:Dobie Gillis:/home/dobie:/bin/bash
```

Пример записей /etc/passwd

- **login** - Символический идентификатор пользователя
- **password** - Поле пароля - не используется
- **UID** - Идентификатор пользовательской УЗ
- **GID** - Идентификатор пользовательской приватной группы
- **GECOS** - Описание/комментарий
- **home** - Домашний каталог пользователя
- **shell** - Командный интерпретатор по-умолчанию

Инструменты для работы с пользовательскими УЗ

- **useradd** - Создаёт новую учётную запись.
- **newusers** - Создание УЗ пользователей в пакетном режиме
- **usermod** - Изменяет данные учётной записи.
- **userdel** - Удаляет существующую учётную запись.
- **chfn** - Изменяет поле GECOS.
- **chsh** - Устанавливает новый командный интерпретатор.

Примеры создания пользовательских УЗ

- Создание пользователей с настройками по умолчанию - домашний каталог, членство в группах, командный интерпретатор

```
$ useradd dexter
```

- Явно указанные параметры при создании пользователя

```
$ useradd -s /bin/csh -m -k /etc/skel -c "Bullwinkle J Moos
G wheel bmoose
```

- Параметры по-умолчанию

`/etc/login.defs`

`/etc/default/useradd`

Раздел 2

БД паролей пользователей

Формат записи о пароле пользователя

`login:hash:LAST_DAY:MIN_DAYS:MAX_DAYS:WARN_DAYS:`
`INACTIVE:EXPIRE_DATE:RESERVED`

- **login - имя пользователя** соответствует /etc/passwd
- **hash - хеш пароля** необратимое криптографическое преобразование
- **LAST_DAY** - Дата последнего изменения пароля
- **MIN_DAYS** - Через сколько дней можно будет поменять пароль
- **MAX_DAYS** - Через сколько дней пароль устареет

- **WARN_DAYS** - За сколько дней до того, как пароль устареет, напомнить о необходимости его смены
- **INACTIVE** - Через сколько дней после того, как пароль устареет, заблокировать учётную запись пользователя
- **EXPIRE_DATE** - Дата блокировки учётной записи
- **RESERVED** - Зарезервированное поле, не используется

Размещение базы данных

БД паролей не должна быть доступна для чтения/записи обычными пользователями. Для изменения пароля предполагается использовать SUID/SGID утилиты, такие как passwd

`/etc/shadow`

- Традиционный для UNIX-подобных систем и ОС на основе ядра Linux вариант хранения паролей

Инструменты работы с БД паролей

- **chage** - Смена временных ограничений УЗ/пароля

```
$ chage -m MIN_DAYS -M MAX_DAYS -I INACTIVE -d LAST_DAY
$ chage -l <user>
```

- **passwd** - Задаёт новый пароль пользователя.
- Суперпользователь меняет пароль другим, пользователь - себе.

Раздел 3

Учетные записи групп

/etc/group

```
group:x:GID:user1,user2,user3
```

- Можно редактировать вручную, но лучше использовать соответствующие инструменты

Управление группами

- **groupadd** - Добавление новой группы
- **groupmod** - Изменение группы и добавление новых пользователей в группу
- **groupdel** - Удаление группы

Управление членством пользователя в группах

- **usermod** - Управление членством пользователя в группах
- **gpasswd** - Управление составом группы

Определение принадлежности пользователя к группам

- **id** - Информация о пользовательских идентификаторах процесса
- **groups** - Список групп для текущего пользователя

```
$ id -Gn [user1, user2 ...]
$ groups [user1, user2 ...]
$ grep <user> /etc/group
$ getent group <grop>
```

- Информация про себя - без параметров. Из под другого пользователя - указываем имя (не требует суперпользователя)

Раздел 4

Информация по пользователям в системе

Пользователи, работающие в системе

- **who** - Пользователи работающие в системе

```
$ who
root         tty2         2013-10-11 10:00
sysadmin     pts/0        2013-10-11 09:59 (:0.0)
sysadmin     pts/1        2013-10-11 10:00 (example.com)
```

- **w** - Пользователи, работающие в системе

```
$ w
 11:44:13 up  3:13,  1 user,  load average: 0,95, 0,89, 0,57
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
user1    tty1      08:31    3:13m  3:06   0.05s /bin/sh
```

Параметр	Описание
LOGIN@	время входа в систему
IDLE	время с момента запуска какой-либо команды
JCPU	время использования CPU с входа в систему
PCPU	время использования CPU текущим процессом
WHAT	процесс, который выполняется

История входов в систему

- **last** - История входов пользователей
- Утилита анализирует файл **/var/log/wtmp** - история логинов пользователей

```
$ last
user1 tty1  :0       Wed Oct 30 08:31   still logged in
root  pts/1 localhost Sun Oct 27 11:54 - 14:35  (02:40)
```

Раздел 5

Повышение привелегий в системе

Выполнение привилегированных операций

Работа под суперпользователем

- Подразумевается, что пользователь **не входит в систему** под учетной записью суперпользователя
- Требования безопасности - не держать интерактивную сессию под суперпользователя, не запускать под суперпользователем пользовательские приложения
- УЗ суперпользователя одна, а администраторов системы, которым она нужна м.б. несколько

Повышение привилегий

- Для выполнения задач администрирования системы, администраторы:
  - работают под **обычными** УЗ пользователей
  - при необходимости выполнения операций, требующих повышенных привелегий в системе **повышают свои полномочия** до соотв. уровня

Механизмы повышения привилегий

- **su (switch user)** - *явная* передача полномочий
- **sudo (switch user do)** - *контролируемая* передача полномочий
- **suid/sgid** - запуск процессов со специальными атрибутами
- **capabilities** - привилегии
- **PolicyKit** - политики предоставления возможности выполнения определенных привилегированных операций пользователям

Команда su

- **повторная регистрация** - УЗ для регистрации указывается в качестве параметра (по умолчанию - root).
- Требуется знать (и ввести) **пароль** УЗ, в которую переходим
- Обычно используется в сеансе командного интерпретатора, в итоге запускается командный интерпретатор в контексте безопасности указанной УЗ

```
su [-|-l] [имя_пользователя [аргумент ... ]]
```

- опция **-l** - запускает командный интерпретатор с параметрами окружения нового пользователя

Команда su - примеры

```
$ su -
Password: ********
$ id
uid=0(root) gid=0(root) groups=0(root)
$ exit
$ su admin
Password: ********
$ exit
```

sudo

- **Контролируемая** передача полномочий - запускаются отдельные программы, а не сеанс
- Могут явно задаваться программы, которые должны быть запущены с повышенными полномочиями
- Перед передачей полномочия у пользователя может запрашиваться пароль своей УЗ
- Все запуски программ через **sudo** журнализируются

```
$ head /etc/shadow
head: cannot open '/etc/shadow' for reading:
  Permission denied
$ sudo head /etc/shadow
[sudo] password for sysadmin: netlab123
...
```

Синтаксис /etc/sudoers

```
пользователь хост = (д_пользователь:группа)опт: команды
```

- **пользователь (или группа)** - кому можно выполнять
- **хост** - на каком узле
- **д_пользователь:группа** - в кого разрешено переходить
- **команды** - что разрешено выполнять

```
root    ALL=(ALL:ALL) ALL
backup  ALL=(root:root) /bin/tar, /usr/bin/rsync
update  ALL=(root:root) /usr/bin/apt-get
%sudo   ALL=(ALL:ALL) ALL
%adm    ALL=(ALL:ALL)NOPASSWD: ALL
#includedir /etc/sudoers.d
```